

Ein multikriterielles Verfahren zum Entwurf zuverlässiger und ökonomischer Netzwerktopologien

Dirk Reichelt und Marc-Christian Knauf
Fachgebiet Wirtschaftsinformatik für Industriebetriebe (WI 1)
Technische Universität Ilmenau
Dirk.Reichelt@tu-ilmenau.de

Dieser Beitrag beschäftigt sich mit dem Aufbau kostengünstiger und zuverlässiger Kommunikationsnetzwerke. Für den Aufbau des Kommunikationsnetzes stehen je Verbindung verschiedene Leitungstypen mit unterschiedlichen Zuverlässigkeiten und Kosten zur Verfügung. Im Rahmen der multikriteriellen Planung ist eine pareto-optimale Lösungsmenge von Netzwerken zu finden. Aufgrund der hohen Komplexität des Problems (NP-vollständig) werden üblicherweise heuristische Optimierungsverfahren zur Lösung eingesetzt. In der Vergangenheit wurden insbesondere Genetische Algorithmen zur Lösung des Problems eingesetzt. Der vorliegende Beitrag stellt eine speziell für die multikriterielle Optimierung angepasste Version des lokalen Suchverfahrens Simulated Annealing für das Problem vor. Experimentelle Untersuchungen der Leistungsfähigkeit des multikriteriellen Simulated Annealing Ansatzes zeigen, dass das Verfahren den Lösungen bisherigen Ansätzen überlegen ist und eine Pareto-Front höherer Güte erstellt.

1 Einleitung

Beim Einsatz von netz- und internetbasierten IT-Anwendungen ist die Zuverlässigkeit und Sicherheit der zugrundeliegenden Kommunikationssysteme einer der Schlüsselfaktoren für den Erfolg einer Geschäftstätigkeit. Insbesondere mit der zunehmenden Integration von IT-Systemen entlang der Wertschöpfungskette wächst die Abhängigkeit von zuverlässigen und störungsfrei operierenden Kommunikationsnetzwerken. Bei der Planung von vernetzten IuK-Systemen ist daher die Zuverlässigkeit der Kommunikationsinfrastruktur auch bei Ausfällen einzelner Netzwerkkomponenten von hoher Bedeutung. Als Maß für die Robustheit eines Netzwerks gegenüber Ausfällen einzelner Leitungen wird die

„All-Terminal“-Zuverlässigkeit verwendet. Dieses Maß gibt die Wahrscheinlichkeit dafür an, dass bei gegebenen Ausfallwahrscheinlichkeiten für einzelne Netzwerkverbindungen sämtliche Knoten des Netzwerks noch miteinander kommunizieren können [3]. Als konfliktäres Ziel steht einer hohen Zuverlässigkeit die Forderung nach Netzwerktopologien mit geringen Kosten gegenüber.

In bisherigen Arbeiten wurde das Problem häufig als Optimierungsproblem mit einer Zielgröße (Kosten oder Zuverlässigkeit) betrachtet [6, 2, 14]. Das zweite Ziel wurde über Nebenbedingungen abgebildet. Mit dieser Arbeit wird ein multikriterielles Entwurfsverfahren vorgestellt, das Kosten und Zuverlässigkeit parallel betrachtet. Hierfür wird das Pareto Simulated Annealing (PSA) Verfahren [5] auf die Problemstellung angepasst. An Stelle einer einzigen Lösung wird durch PSA eine Menge an Lösungen generiert, die dem Entscheider eine Auswahl entsprechend individueller Präferenzen ermöglicht. Experimentelle Ergebnisse zeigen, dass die Qualität der mit PSA erzielten Lösungen der eines multikriteriellen Genetischen Algorithmus überlegen ist.

Der Beitrag ist wie folgt gegliedert: Kapitel 2 stellt das Netzwerkdesignproblem (NDP) als multikriterielles Optimierungsproblem vor. Das PSA-Verfahren sowie eine geeignete Nachbarschaftssuche für die Problemstellung wird in Abschnitt 3 vorgestellt. Die mittels PSA erzielten Ergebnisse sowie ein Vergleich zu einem multikriteriellen Genetischen Algorithmus finden sich in Abschnitt 4. Der Beitrag schließt mit einer Zusammenfassung.

2 Netzwerktopologieplanung als multikriterielles Optimierungsproblem

Bei der Planung von Kommunikationsnetzwerken müssen eine Reihe unterschiedlicher Kriterien wie Zuverlässigkeit, Installationskosten, Betriebskosten, Durchsatz, Kapazitäten oder maximale Laufzeiten berücksichtigt werden. An Hand der unterschiedlichen Kriterien wird deutlich, dass unterschiedlicher zueinander konfliktäre Zielsetzungen, in dem NDP zu berücksichtigen sind. In der Vergangenheit wurde das Problem häufig als unikriterielles Problem betrachtet. Smith u. a. [7, 8, 6] stellen Genetische Algorithmen vor, bei denen die Netzwerkkosten minimiert werden und die Zuverlässigkeitsnebenbedingung über einen Strafterm direkt in die Zielfunktion integriert wird. Reichelt u. a. [14] verwenden einen Genetischen Algorithmus, der die Netzwerkkosten minimiert und die Erfüllung einer Zuverlässigkeitsnebenbedingung mittels einer Reparaturfunktion sicherstellt. In [2] wird ein Simulated Annealing Ansatz vorgestellt, der die Netzwerkkosten minimiert und die Zuverlässigkeit über eine Nebenbedingung abbildet. Ein Genetischer Algorithmus für das Netzwerkdesign mit mehreren Zuverlässigkeitszielen wird in [13] vorgestellt. Als Zuverlässigkeitsmaß wird die „k-Terminal“-Zuverlässigkeit verwendet. Eine Betrachtung von Zuverlässigkeit und Kosten als multikriterielles Optimierungsproblem erfolgt in [9, 1]. In beiden Arbeiten untersuchen die Autoren das NDP mit Hilfe verschiedener multikriterieller Genetischer Algorithmen.

Im vorliegenden Beitrag soll das NDP als multikriterielles Optimierungsproblem mit den Zielgrößen Kosten und Zuverlässigkeit betrachtet werden. Die zu verbindenden Knoten sowie die zur Verfügung stehenden Kanten werden als ungerichteter Graph $G(K, E)$ mo-

delliert. K ist die Menge der Knoten und E die Menge der Kanten im Graphen. Durch jede Kante bzw. jedem Knoten wird eine Verbindung bzw. ein Knoten des zugehörigen Netzwerks repräsentiert. Es wird davon ausgegangen, dass die Positionen der Knoten fest vorgegeben sind und deren Installationskosten für die Planung nicht relevant sind. Für jede Kante $e_{ij} \in E$ zwischen den Knoten i und j besteht die Auswahlmöglichkeit zwischen verschiedenen Zuverlässigkeitsoptionen l_k ($k = 0 \dots n$) welche sich hinsichtlich Kosten $c(l_k(e_{ij}))$ und Ausfallsicherheit $r(l_k(e_{ij}))$, unterscheiden. Dabei entspricht $l_k(e_{ij})$ der aktuell für die Kante e_{ij} gewählten Option. Weiterhin gelten die Annahmen, dass die Knoten zuverlässig arbeiten, jede Kante sich entweder im Zustand $s_{e_{ij}}$ „operational“ ($s_{e_{ij}} = 1$) oder „failed“ ($s_{e_{ij}} = 0$) befindet, die Ausfallwahrscheinlichkeiten der einzelnen Leitungen unabhängig voneinander sind, die Verbindungen bidirektional sind und Reparaturen ausgefallener Verbindungen nicht berücksichtigt werden. Eine Lösung x (ein Netzwerk N des Optimierungsproblems) wird durch den Subgraph $G_N(K, E_N \subset E)$ dargestellt. Wir definieren $C(x)$ als die Gesamtkosten eines Netzwerks, welche sich aus den Kosten $c_k(l(e_{ij}))$ der aktuell gewählten Optionen für die Kanten $e_{ij} \in G_N$ zusammensetzen. Als Zuverlässigkeitsmaß wird die „All-Terminal“-Zuverlässigkeit $R_{All}(x)$ verwendet. Dieses Maß berechnet sich aus den vorgegebenen Zuverlässigkeiten der einzelnen Leitungen, welche im Netzwerk eingesetzt werden, und gibt die Wahrscheinlichkeit dafür an, dass sämtliche Knoten eines Netzwerks miteinander kommunizieren können. Verfahren zur Berechnung der „All-Terminal“-Zuverlässigkeit werden [4, 3, 11] beschrieben. Als multikriterielles Optimierungsproblem wird das hier betrachtete NDP wie folgt dargestellt:

$$\begin{aligned} \min f_1(x) &= C(x) \\ \max f_2(x) &= R_{All}(x) \\ \text{mit: } R_{All}(x) &> 0 \end{aligned} \tag{1}$$

Die Menge sämtliche gültiger Lösungen für das Problem ist D . Eine Lösung $x \in D$ ist nicht dominiert (pareto-optimal), wenn gilt:

$$\nexists x' \in D \text{ mit } C(x') \leq C(x) \wedge R_{All}(x') > R_{All}(x) \tag{2}$$

Ergebnis eines multikriteriellen Optimierungsverfahrens ist nicht eine einzelne Lösung, sondern eine Menge von nicht dominierten Lösungen Y_{known} . Sämtliche nicht dominierten Lösungen für ein Problem bilden die Pareto-Front Y_{true} . Bei der Auswahl einer optimalen Lösung $x_{opt} \in Y_{true}$ muss der Entscheidungsträger eigene Präferenzen λ für die Zielfunktionen f_j festlegen. Wir bezeichnen eine nicht dominierte Lösung $x \in Y_{true}$ als unterstützt, wenn es einen Vektor von nicht negativen Gewichten $\lambda = [\lambda_1, \dots, \lambda_j]$ gibt, so dass x ein Optimum des Problems ist. Für eine unterstützte Lösung gilt:

$$\begin{aligned} \max \sum_{j=1}^J \lambda_j f_j(x) \\ \text{mit : } x \in D \text{ und } \sum_j \lambda_j = 1 \end{aligned} \tag{3}$$

3 Ein multikriterielles Simulated Annealing Verfahren für den Netzwerktopologieentwurf

3.1 Der Pareto Simulated Annealing Ansatz

Die Metaheuristik Simulated Annealing ist ein der Physik entlehntes Verfahren. Es ist in Analogie zu dem in der Natur vorkommenden Kristallisierungsprozess beim Abkühlen von Metallen und Flüssigkeiten entwickelt worden. Hat eine Schmelze eine relativ hohe Temperatur, bewegen sich die Teilchen darin fast frei umher, ohne sich gegenseitig stark zu beeinflussen. Wenn nun die Abkühlung hinreichend langsam erfolgt, so stellt sich bei jeder Temperatur ein thermisches Gleichgewicht ein. Die Substanz gelangt dann in eine Grundstruktur mit minimalem Energieniveau z.B. ein Kristall. Diese Struktur stellt das Energieminimum für den jeweiligen Stoff dar. Wichtig ist dabei, dass die Abkühlung nicht zu schnell erfolgt denn sonst kommt es zum sogenannten Quenching. Dabei entstehen nicht optimale Strukturen, die ein höheres Energieniveau aufweisen. Der Versuch dieses Phänomen auf einem Computer zu übertragen, und zur Lösung kombinatorischer Optimierungsprobleme zu nutzen, wurde erstmals von Kirkpatrick u. a. [12] unternommen. Ausgehend von einer Startkonfiguration $s_0 \in D$ wird ein Element aus der Nachbarschaft $s_1 \in NS(s_0)$ gewählt und die Differenz der Zielfunktionswerte $\Delta f = f(s_1) - f(s_0)$ berechnet. Ist der Zielfunktionswert der Nachbarkonfiguration besser ($\Delta f \leq 0$ bei Minimierung des Zielfunktionswertes), wird diese Lösung als Ausgangspunkt für die weitere Suche benutzt. Mit einer Wahrscheinlichkeit $P(T)$ werden auch schlechtere Lösungen ($\Delta f > 0$), die ein bestimmtes Akzeptanzkriterium erfüllen, als Basis für die weitere Suche verwendet. Der Temperaturparameter T , der sich während der Optimierung langsam verringert, steuert die Akzeptanzwahrscheinlichkeit für schlechte Lösungen. Je weiter die Abkühlung fortgeschritten ist, desto geringer wird die Akzeptanzwahrscheinlichkeit für schlechtere Lösungen.

In Anlehnung an das Simulated Annealing Verfahren für Optimierungsprobleme mit einer Zielgröße wurden verschiedene Simulated Annealing Verfahren für multikriterielle Optimierungsprobleme entwickelt. In [15] stellt Serafini das Simulated Annealing for Multi-Objective Optimization (SMOSA) vor. In den Grundzügen ähnelt das Verfahren sehr stark dem prinzipiellen Aufbau eines Simulated Annealing Ansatzes nach [12]. An Stelle einer einzigen optimalen Lösung erzeugt das Verfahren als Ergebnis eine Menge nicht dominierter Lösungen Y . Mit der Arbeit von Ulungu [10] wird ein Simulated Annealing (MOSA) Ansatz für multikriterielle Optimierungsprobleme vorgestellt, der sich Prinzipien einer populationsbasierten Optimierung bedient. Im Gegensatz zu SMOSA nutzt MOSA nicht nur eine Lösung während der Optimierung, sondern setzt parallel mehrere verschiedene Lösungen ein, um den Problemraum besser abzudecken und eine möglichst große Menge an nicht dominierten Lösungen zu erstellen. Aufbauend auf den Arbeiten von Serafini und Ulungu entwickelten Czyzak und Jaszkievicz das PSA-Verfahren, das in dem vorliegenden Beitrag genutzt wird. PSA nutzt eine Menge (Population) von interagierenden Lösungen. PSA greift dabei die Idee von [15] zur Kontrolle der Kriteriengewichte mittels multikriterieller Regeln für die Akzeptanzwahrscheinlichkeit auf. Ziel ist es, eine Streuung der generierten Lösungen über die ganze Menge von nicht dominierten

Lösungen zu sichern. Je höher die Gewichtung eines gegebenen Kriteriums, umso geringer ist die Wahrscheinlichkeit, dass eine neue Lösung akzeptiert wird, die den Wert des Kriteriums vermindern und umso größer ist die Wahrscheinlichkeit der Verbesserung des Wertes des Kriteriums. Durch Kontrolle der Gewichte kann man die Wahrscheinlichkeit für Verbesserung der Werte eines Kriteriums erhöhen oder vermindern. Der komplette Ablauf von PSA wird im Folgenden beschrieben.

Erzeuge eine Menge $S_0 \subset D$

$Y_{known} = \emptyset$

$\forall_{x \in S_0}$ update Y_{known}

$T = T_0, S = S_0$

wiederhole

$\forall_{x \in S}$ do

Erstelle $y \in NS(x)$

if y ist nicht dominiert von x do

update Y_{known} mit y

wähle die Lösung $x' \in S$ nächstmöglich zu x und nichtdominiert von x

wenn es keine Lösung x' gibt oder es ist die erste Iteration mit x do

Setze λ_j zufällig, so dass gilt

$\forall_j \lambda_j \geq 0$ und $\sum_j \lambda_j = 1$

sonst

$\forall f_j$ do

$$\lambda_j = \begin{cases} \alpha \lambda_j^x & \text{wenn } f_j(x) \geq f_j(x') \\ \frac{\lambda_j^x}{\alpha} & \text{wenn } f_j(x) < f_j(x') \end{cases}$$

normalisiere λ so dass $\sum_j \lambda_j = 1$

setze $x = y$ mit $P(x, y, T, \lambda)$

wenn Bedingung für Temperatursenkung erfüllt, so vermindere T

solange (Abbruchkriterium nicht erfüllt)

Der Faktor α ist eine Konstante nahe eins (i.a. 1,05) zur Anpassung der Gewichte. Der Vektor $\lambda^x = [\lambda_1^x, \dots, \lambda_j^x]$ repräsentiert die im letzten Durchlauf verwendete Gewichtung der einzelnen Zielfunktionen $f_j(x)$. Unter der Zielsetzung am Ende des Verfahrens mit der besten bekannten Lösungsmenge Y_{known} eine möglichst große Menge aus der optimalen Lösungsmenge Y_{true} abzudecken, wird der Vektor mit den Gewichten λ für die einzelnen Zielfunktionen bei jedem Durchlauf so angepasst, dass alle Gewichte der Zielfunktionen bei denen $f_j(x) \geq f_j(x')$ größer werden und alle anderen Gewichte verringert werden. Auf diese Weise stellt PSA sicher, dass in jedem Durchlauf der Abstand der Lösungen Menge Y_{known} zunimmt und somit unterschiedliche Regionen des Suchraums D durch die verschiedenen Lösungen $x \in Y_{known}$ durchsucht werden. Für jedes Element aus Y_{known} wird in einem Durchlauf eine benachbarte Lösung y gesucht. Über die Funktion $P(x, y, T, \lambda)$ wird die Akzeptanzwahrscheinlichkeit für eine Lösung y bestimmt. Eine Lösung $y \in D$ wird von der Prozedur *Update* in Y_{known} eingefügt, wenn es keine Lösung $x \in Y_{known}$ gibt, die y dominiert. Gleichzeitig löscht *Update* beim Einfügen einer neuen pareto-optimalen

Lösungen in Y_{known} sämtliche durch diese Lösung dominierten Lösungen.

3.2 Adaption des PSA-Verfahrens für das Netzwerk Design Problem

Für den Einsatz des PSA-Verfahrens für das NDP muss eine Funktion $NS(x)$ zur Erzeugung einer benachbarten Lösung für x entworfen werden. Eine Nachbarschaftslösung $NS(x)$ wurde nach drei gleichverteilten möglichen Schritten generiert. Es wurde entweder:

- eine Kante aus dem Netzwerk gelöscht ($E_N \setminus \{e_{ij}\}$ mit $e_{ij} \in E_N$)
- eine Kante dem Netzwerk hinzugefügt ($E_N \cup \{e_{ij}\}$ mit $e_{ij} \in E, e_{ij} \notin E_N$)
- oder die Option l_k einer Kante des Netzwerks verändert ($l_{k+1}(e_{ij})$ mit $e_{ij} \in E_N$)

4 Experimentelle Ergebnisse

4.1 Experimentelles Design

Als Testproblem wurde das in [6] vorgestellte Netzwerk verwendet. Das Netzwerk ist die vereinfachte Version eines realen Netzwerk Design Problems von 19 Universitäten und Forschungseinrichtungen in neun Städten der Türkei. Als Verbindung zwischen jedem Knotenpaar werden drei mögliche Arten von Glasfaserverbindungen verwendet, deren Kosten und Zuverlässigkeiten gegeben sind mit 333\$/km 96%, 433\$/km 97.5% und 583\$/km 99%. Die Implementierung des PSA-Verfahrens erfolgt mit Hilfe der MOM-HLib++. Die Initialisierungsmethode erstellt zufallsbasierte 2-fach kantenverbundene Netzwerke. In jedem Durchlauf wurde nach einer gegebenen Anzahl an Bewegungen die Temperatur gesenkt. Der Abkühlungsfaktor der Temperatur wurde auf 0,9 gesetzt. Die Starttemperatur lag bei 50. Der Algorithmus terminiert, nachdem eine Temperatur 0,5 erreicht wurde. Das Verfahren wurde mit einer Größe von 8, 16 und 32 parallel Lösungen getestet. Für jede Anzahl an parallelen Lösungen wurden Versuche mit insgesamt 128, 256 bzw. 512 Bewegungen pro Temperaturlevel durchgeführt. Dies bedeutet, je größer die Anzahl der parallel untersuchten Lösungen ist, desto weniger Bewegungen wurden für jede einzelne Lösung durchgeführt. Bei einer einzelnen Lösung würden z.B. 512 Bewegungen pro Temperaturlevel durchgeführt, bei 16 Lösungen waren es noch 32 Bewegungen für jede einzelne Lösung. Für die Bestimmung der „All-Terminal“-Zuverlässigkeit $R_{All}(x)$ wurde eine einfache Monte Carlo Simulation [11] mit jeweils 100000 Stichproben verwendet. Im Gegensatz zu [9, 1], wo lediglich mit 10000 Stichproben gearbeitet wurde, steigt so die Wahrscheinlichkeit, mit Hilfe der Monte Carlo Schätzung die exakte Zuverlässigkeit eines Netzwerkes zu ermitteln (vgl. [11]).

4.2 Maße für den Vergleich multikriterieller Optimierungsverfahren

Ziel einer multikriteriellen Metaheuristiken ist es, mit der generierten Lösungsmenge Y_{known} eine gute Annäherung an eine als effizient betrachtete Menge an Lösungen Y_{true} zu finden. Deshalb sind Verfahren und Gütemaße notwendig, die einen Vergleich der

verschiedenen (nicht-dominierten) Lösungen ermöglichen. Im Folgenden werden die in diesem Beitrag verwendeten Maße vorgestellt.

Zur Bewertung der einzelnen Verfahren werden die folgende Kennzahlen aus [16] herangezogen:

1. Overall Non-dominated Vector Generation (ONVG); zählt die Anzahl der Lösungen in der generierten Pareto-Front Y_{known}

$$ONVG \triangleq |Y_{known}|_c \quad (4)$$

wobei $||_c$ die Kardinalität der Menge Y_{known} bezeichnet.

2. Overall True Non-dominated Vector Generation (OTNVG); zählt die Anzahl der Lösungen in der Pareto-Front Y_{known} welche auch in der "wahren" Pareto-Front Y_{true} vorhanden sind.

$$OTNVG = |\{y|y \in Y_{known} \wedge y \in Y_{true}\}|_c \quad (5)$$

3. Overall Non-dominated Vector Generation Ratio (ONVGR);

$$ONVGR = \frac{ONVG}{Y_{true}} \quad (6)$$

bezeichnet das Verhältnis zwischen der Anzahl der Lösungen in Y_{known} zu der Anzahl der Lösung in der Pareto-Front Y_{true} . Da es das Ziel ist, mit der Menge Y_{known} nah an der wahren Pareto-Front Y_{true} zu sein, sind Werte möglichst nah an eins positiv.

4. Error Ratio (E)

$$E = \frac{\sum_{x \in Y_{known}} e_x}{ONVG} \text{ mit:} \quad (7)$$

$$e_x = 0 \text{ wenn } x \in Y_{known} \wedge x \in Y_{true}$$

$$e_x = 1 \text{ sonst}$$

Es spiegelt das Verhältnis zwischen Lösungen in der Menge Y_{known} wider, welche keine Mitglieder von Y_{true} sind. Ein Error Ratio (E) nahe eins bedeutet einen schwachen Zusammenhang zwischen erzeugter und wahrer Pareto-Front Werte nahe Null sind gewünscht.

5. Generational Distance (G)

$$G = \frac{\sqrt{\sum_{x \in Y_{known}} d_x^2}}{ONVG} \quad (8)$$

wobei d_x der Abstand (im Lösungsraum) zwischen jedem $x \in Y_{known}$ und seinem nächsten zugehörigen Nachbarn in der Pareto-Front Y_{true} bedeutet. Eine hoher Wert für G bedeutet, dass Y_{known} weit entfernt von Y_{true} ist, ein $G = 0$ ist die ideale Situation.

Vom Betrachter wird als natürlichste Kennzahl im allgemeinen OTNVG angenommen bzw. $\frac{OTNVG}{Y_{true}} \cdot 100$ also der Prozentsatz an gefundenen Lösungen in der Referenzmenge. Dieses Maß hat aber einige Nachteile, da es bei einer Vielzahl von Lösungen in Y_{true} fast unmöglich ist einen signifikanten Prozentsatz an effizienten Lösungen in einer annehmbaren Zeit zu finden. Dies ist meist auch gar nicht erforderlich, viel wichtiger ist es, in einer kurzen Zeit möglichst nah an die Lösungsmenge Y_{true} zu kommen. Distanz-Kennzahlen wie G besitzen eine wesentlich höhere Aussagekraft. Sie kann Aufschluss darüber geben, wie weit die erzeugte Lösungsmenge Y_{known} von der Lösungsmenge Y_{true} entfernt ist. Eine gute Annäherung an die Lösungsmenge Y_{true} wird durch eine Aggregation aus allen erzeugten Lösungsmenge erreicht [9, 1].

Als zweite Bewertungsmethode soll die Verteilung der Lösungen gegenüber einer Referenzlösung R aus [9] bewertet werden. Als Bewertungsgrundlage nutzen wir das Kennzahlensystem von Jazzkiewicz [5], nach dem eine gute Annäherung der Lösungsmenge Y_{known} an R demnach dann erreicht ist, wenn es für jede Lösung $y \in R$ eine nahe Lösung $x \in Y_{known}$ gibt. Anders als G wird also die Distanz eines Elements der Referenzlösung zur gefundenen Lösungsmenge und nicht umgekehrt bewertet. Zu diesem Zweck hat er zwei Distanz-Kennzahlen aufgestellt.

$$Dist1 = \frac{1}{R} \sum_{y \in R} \{ \min_{x \in Y_{known}} \{d(x, y)\} \} \quad (9)$$

$$Dist2 = \max_{y \in R} \{ \min_{x \in Y_{known}} \{d(x, y)\} \} \quad (10)$$

wobei $d(x, y)$ die Distanz zwischen zwei Lösungen darstellt.

Dist1 sagt etwas über die durchschnittliche Distanz von $y \in R$ zur nächsten Lösung in Y_{known} aus. Dist2 dagegen betrachtet den schlimmsten Fall. Über das Verhältnis zwischen

$$\frac{Dist2}{Dist1} \quad (11)$$

lassen sich Aussagen über die Gleichförmigkeit der Verteilung treffen. Je kleiner die Werte dieser Maße desto besser ist die Annäherung der Menge Y_{known} an R .

Für die Distanz $d(x, y)$ wird in diesem Beitrag

$$d(x, y) = \left(\sum_{j=1}^J (w_j (f_j(x) - f_j(y)))^2 \right)^{1/2} \quad (12)$$

verwendet. Dabei sind w_j , $j = 1, \dots, J$ die Gewichte zur Normalisierung der Kriterien f_j , die die Spannweite des jeweiligen Kriteriums im Lösungsraum D darstellen.

4.3 Ergebnisse

Für die Auswertung werden die durch PSA generierten Lösungsmengen Y_{known} mit der in [9] angegebenen Referenzlösungsmenge R und der durch den Einsatz von MOSA, SMOSA und PSA gewonnenen Menge Y_{true} , die sich aus 80 pareto-optimalen Lösungen zusammensetzt, verglichen.

Tabelle 1 zeigt die Auswertung der mittels PSA erzielten Ergebnisse mit Hilfe der Kennzahlen aus [9, 1]. Dabei werden die durch das Verfahren generierten Lösungsmengen der Lösungsmenge Y_{true} gegenübergestellt. P gibt die Anzahl der parallel betrachteten Lösungen (Population) an, NM ist die Gesamtanzahl von Bewegungen (Neighborhood Moves) pro Temperaturschritt aller Elemente der Population und t_{run} zeigt die für den Lauf benötigte Zeit in Sekunden. Der Wert für $ONVG$ zeigt, dass mit steigender Anzahl

	P	NM	$ONVG$	$OTNVG$	$ONVGR$	E	G	t_{run}
1	8	128	46	0	0,5750	1,0000	0,0243	5155
2	16	128	32	0	0,4000	1,0000	0,0270	5255
3	32	128	38	0	0,4750	1,0000	0,0385	5279
4	8	256	76	25	0,9500	0,6711	0,0116	9832
5	16	256	51	0	0,6375	1,0000	0,0148	9821
6	32	256	38	0	0,4750	1,0000	0,0305	10527
7	8	512	96	26	1,2000	0,7292	0,0054	20048
8	16	512	64	3	0,8000	0,9531	0,0110	20620
9	32	512	43	0	0,5375	1,0000	0,0180	19916

Tabelle 1: Ergebnisse und Auswertung PSA-Läufe

an Bewegungen pro Temperaturlevel (NM) die Anzahl der durch das Verfahren generierten nicht dominierten Lösungen ($ONVG$) zunimmt. Dies lässt sich damit begründen, dass für jede einzelne Lösung pro Temperaturschritt mehr Nachbarschaftslösungen untersucht werden. Wie die Tabelle zeigt, wurde in allen durchgeführten Experimenten die höchste Anzahl an pareto-optimalen Lösungen ($ONVG$) bei gleichem NM jeweils mit einer Populationsgröße (P) von 8 erzielt. Wie die Auswertung mittels $OTNVG$ Maß zeigt, wird bei einer Populationsgröße von 8 ebenfalls die größte Anzahl an nicht dominierten Lösungen in Y_{true} generiert. Stellt man für die Populationsgröße von 8 die Laufzeit t_{run} der Ergebnisgüte ($ONVG$ und $OTNVG$) gegenüber, so ist festzustellen, dass durch Verdoppelung der Laufzeit (Zeile 4 zu 7) lediglich ein geringer Zuwachs an zusätzlichen pareto-optimalen Lösungen entsteht. Die Distanz G zeigt, dass mit einer höheren Anzahl an lokalen Suchschritten (NM) eine Distanz zwischen der Pareto-Front Y_{true} und der durch das Verfahren erzeugten Pareto-Front Y_{known} geringer wird. Die geringste Abweichung von Y_{known} und Y_{true} konnte mit der Parameterkombination $P = 8$ und $NM = 512$ erreicht werden.

Die Abbildungen 1(a) und 1(b) zeigen einen Vergleich der Pareto-Front Y_{true} sowie der Pareto-Front Y_{known} in der Parameterkombination $P = 8$ und $NM = 512$ bzw. $P = 8$ und $NM = 256$. Wie beide Abbildungen zeigen, deckt die durch das PSA-Verfahren generierte Pareto-Front jeweils einen großen Bereich der Pareto-Front Y_{true} ab. Wäh-

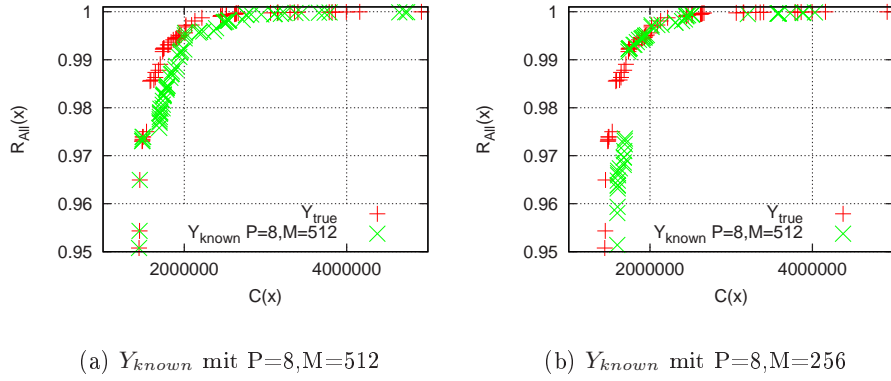


Abbildung 1: Vergleich von Y_{true} und Y_{known}

	$Dist1$	$Dist2$	$Dist2/Dist1$
PSA	0,065	0,595	9,145

Tabelle 2: Distanzen aggregierter Y_{known} (für alle PSA-Läufe) und R

rend die Pareto-Front in der Parameterkombination $P = 8$ und $NM = 256$ eine Vielzahl von Lösungen zu Y_{true} für $R_{All}(x) > 0.975$ beiträgt, so liegen die Lösungen aus $x \in Y_{known}$ mit $x \in Y_{true}$ für die Parameterkombination $P = 8$ und $NM = 512$ im Bereich $R_{All}(x) > 0.975$ und $R_{All}(x) \approx 1$. Für beide Parameterkombinationen kann festgehalten werden, dass die durch PSA gefundenen Fronten sehr nah an der „wahren“ Pareto-Front Y_{true} liegen. Experimentelle Studien mit den Verfahren MOSA, SMOSA und PSA haben gezeigt, dass mit Hilfe des PSA-Verfahrens die besten Lösungen für die hier betrachtete Problemstellung erzielen lassen. Dies zeigt sich auch an Hand der hier präsentierten Ergebnisse. Durch PSA werden 67% der pareto-optimalen Lösungen aus der Pareto-Front Y_{true} gefunden.

Tabelle 2 zeigt die Auswertung mit Hilfe der Distanzmaße aus [5]. Hierfür wird die aggregierte Pareto-Front aller PSA-Läufe mit der Referenzmenge R verglichen. An Hand des $Dist1$ Wertes in Tabelle 2 erkennt man, welche Gleichförmigkeit zwischen der Referenzlösung und der erzeugten Pareto-Fronten besteht. Im Mittel ($Dist1$) ist die durch die mittels der PSA-Läufe erzeugte Pareto-Front nah an der Referenzmenge R . Daraus folgt, dass viele Lösung der Referenzmenge durch die mittels PSA erzeugte Pareto-Front repräsentiert werden. Der maximale Abstand ist demgegenüber relativ hoch, das heißt, dass Lösungen der Referenzmenge überhaupt keine Repräsentation in der Pareto-Front besitzen. Aus den großen Unterschieden der beiden Kennzahlen $Dist1$ und $Dist2$ resultiert der hohe Wert für das Verhältnis zwischen $Dist2/Dist1$. Dies bedeutet eine relativ schlechte Gleichförmigkeit der Pareto-Front gegenüber R . Dies ergibt sich, da die Lösungen in R für $R_{all} < 0.87$ keine Repräsentation in Y_{known} haben und R nur Lösungen mit $C(x) < 3900000$ enthält. Bessere Lösungen in Y_{known} relativieren dies nur bedingte, da die verwendeten Maße eigentlich von schlechteren Werten in Y_{known} gegenüber

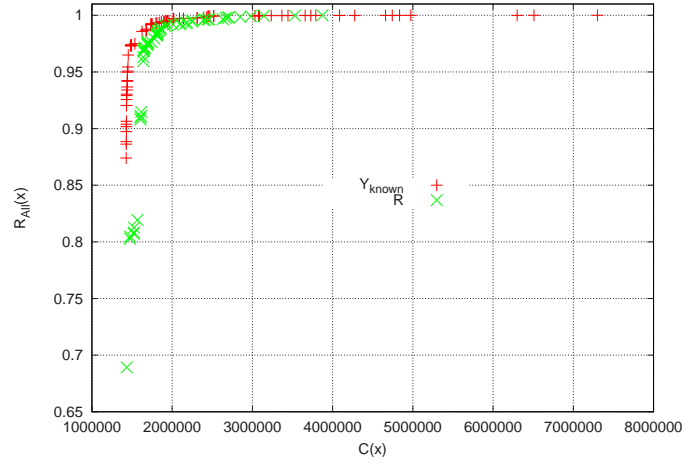


Abbildung 2: Vergleich der Pareto-Fronten R und Y_{known} (alle PSA-Läufe)

der Referenzmenge R ausgehen. Abbildung 2 stellt die Referenzmenge R der Lösungen aus [9] und die aus allen PSA-Läufen generierte Menge an nicht dominierten Lösungen Y_{known} gegenüber. Der Verlauf der beiden Pareto-Fronten zeigt, dass mittels PSA eine deutliche bessere Lösungsmenge generiert wird, deren Elemente viele Lösungen in der Referenzmenge R dominieren. Beim Vergleich beider Lösungsmengen muss zusätzlich beachtet werden, dass Baranů.a. lediglich 10000 Stichproben pro Monte Carlo Simulation durchführen, während im vorliegenden Beitrag 100000 Stichproben für jede Monte Carlo Schätzung verwendet wurden. Für die Varianz des Schätzers $\hat{R}_{All}(G)$ einer einfachen Monte Carlo Simulation gilt $var(\hat{R}_{All}(G)) = \frac{R_{All}(G)(1-R_{All}(G))}{i}$. Daraus folgt, dass die Genauigkeit der Schätzung mit einer größeren Stichprobenanzahl zunimmt und die hier durchgeführte Schätzung eine höhere Genauigkeit besitzt.

5 Zusammenfassung

Dieser Beitrag beschäftigt sich mit der multikriteriellen Planung von zuverlässigen und ökonomischen Netzwerktopologien. Im Gegensatz zu bisherigen Arbeiten, in denen Kosten und Zuverlässigkeit häufig getrennt voneinander betrachtet wurden, findet hier eine simultane Betrachtung beider Zielgrößen statt. Hierfür wurde die Pareto Simulated Annealing Metaheuristik mit einem für die Problemstellung angepassten Nachbarschaftskonzept vorgestellt. An Stelle einer einzigen Lösung generiert das Verfahren eine Menge von zueinander pareto-optimalen Lösungen. Dabei obliegt es dem Entscheidungsträger/Planer entsprechend seinen persönlichen Präferenzen eine Lösung auszuwählen.

In einer experimentellen Studie wurde das PSA für das Netzwerk Design Problem mit unterschiedlichen Populationsgrößen und einer variierenden Anzahl von lokalen Suchschritten getestet. Die experimentellen Ergebnisse zeigen, dass die mittels PSA erzeugte Pareto-Front eine höhere Lösungsgüte als bisher bekannte Lösungen multikriterieller Genetischer Algorithmen aufweist. In einer durch mehrere multikriterielle Simulated An-

nealing Verfahren erstellten Pareto-Front, welche die bisher bekannte Pareto-Front dominiert, generiert PSA 67% der pareto-optimalen Lösungen und zeichnet sich durch eine breite Streuung der Lösungen über den kompletten Problemraum aus.

Literatur

- [1] BARAN, B., S. DUARTE und D. BENÍTEZ: *Telecommunication Network Design with Parallel Multi-objective Evolutionary Algorithms*. In: *IFIP/ACM Latin America Networking Conference*, La Paz, Bolivia, 2003.
- [2] B.DENGIZ und C. ALABAP: *A Simulated Annealing Algorithm for Design of Computer Communication Networks*. In: *Proceedings of World Multiconference on Systems, Cybernetics and Informatics, SCI 2001*, Band 5, 2001.
- [3] CHEN, YUNBIN, JIANDONG LI und JIAMO CHEN: *A New Algorithm For Network Probabilistic Connectivity*. In: *IEEE military communication conference*, 1999.
- [4] COLBOURN, CHARLES J.: *The Combinatorics of Network Reliability*. Oxford University Press, 1987.
- [5] CZYZAK, PIOTR und ANDRZEJ JASZKIEWICZ: *Pareto Simulated Annealing - A Metaheuristic Technique for Multiple-Objective Combinatorial Optimazation*. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.
- [6] DARREN, D., L. ALICE und E. SMITH: *Economic design of reliable networks*. *IIE Transactions*, 30:1161–1174, 1998.
- [7] DENGIZ, B., F. ALTIPARMAK und A. E. SMITH: *A genetic algorithm approach to optimal topological design of all terminal networks*. In: *Intelligent Engineering Systems Through Artificial Neural Networks, Volume 5*, Seiten 405–410. ASME Press, 1995.
- [8] DENGIZ, B., F. ALTIPARMAK und A. E. SMITH: *Local Search Genetic Algorithm for Optimal Design of Reliable Networks*. *IEEE Trans. on Evolutionary Computation*, 1(3):179–188, 1997.
- [9] DUARTE, S. und B. BARÁN: *Multiobjective Network Design Optimisation Using Parallel Evolutionary Algorithms*. In: *Proccedings of XXVII Conferencia Latinoamericana de Informática CLEI'2001*, Merida, Venezuela, 2001.
- [10] E.L., ULUNGU, TEGHEM J., FRONTTEMPS PH. und TUYTTENS D.: *MOSA method: a tool for solving multiobjective combinatorial optimization problems*. *Journal of Multi-Criteria Decision Analysis*, 8:221–236, 1999.
- [11] FISHMAN, G.S.: *Monte Carlo Simulation*. Springer, 1995.
- [12] KIRKPATRICK, S., C. GELATT und M. VECCHI: *Optimization by simulated annealing*. *Science*, 220:671–680, 1983.

- [13] LIU, BAODING und K. IWAMURA: *Topological Optimization Model for Communication Network with Multiple Reliability Goals*. Computer and Mathematics with Applications, 39:59–69, 2000.
- [14] REICHELT, DIRK, FRANZ ROTHLAUF und PETER GMILKOWSKY: *Designing Reliable Communication Networks with a Genetic Algorithm Using a Repair Heuristic*. In: *Proceedings 4th European Conference, EvoCOP 2004*, Band 3004 der Reihe *Lecture Notes in Computer Science*, Seiten 177–187. Springer, 2004.
- [15] SERAFINI, P.: *Simulated Annealing for multiple objective optimization problems*. Multiple Criteria Decision Making. Expand and Enrich the Domains of Thinking and Application, Seiten 283–292, 1994.
- [16] VELDHUIZEN, DAVID A. VAN: *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Doktorarbeit, Air Force Institute of Technology, Department of Electrical and Computer Engineering, Ohio, 1999.